

Einführung in Visual Computing

186.822

eine gemeinsame Lehrveranstaltung von

Werner Purgathofer – Computergraphik

Robert Sablatnig – Computer Vision



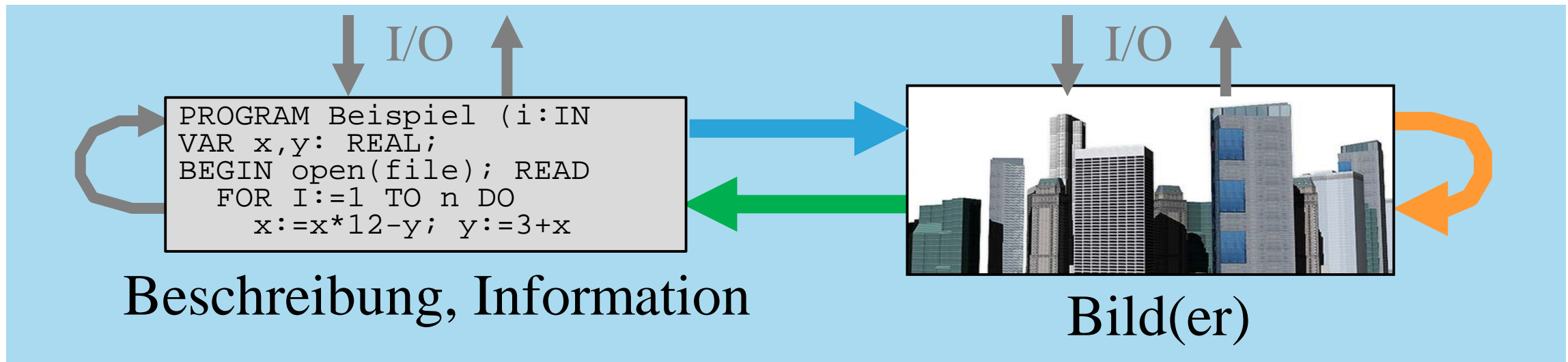
- **Pflicht** in den Bachelorstudien
 - *Medieninformatik & Visual Computing*
 - *Software & Information Engineering*
 - *Medizinische Informatik*
- **Freifach** für alle anderen
- Vertiefende Übungen im Winter-Semester:
 - Einführung in die Computergraphik UE
 - Einführung in die digitale Bildverarbeitung UE

(**Pflicht** nur für *Medieninformatik & Visual Computing*)



Datenverarbeitung mit Bildern

- **Computergraphik:** Beschreibung → Bilder
- **Bildverarbeitung:** “schlechtes” → “besseres” Bild
- **Computer Vision:** Bilder → Beschreibung



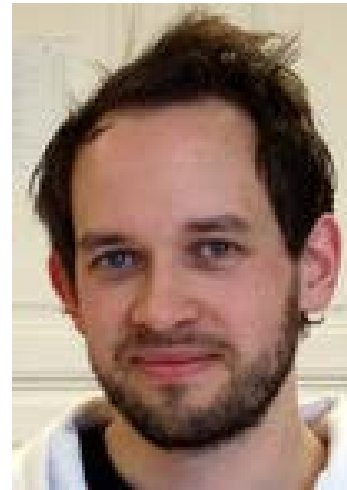
$$CG + BV + CV = „Visual Computing“$$



■ www.cg.tuwien.ac.at/courses/EinfVisComp

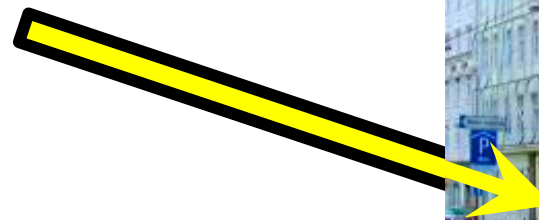
■ **zuständige Assistenten:**

- Johannes Unterguggenberger
- Bernhard Steiner
- Sebastian Zambanini
- evc@cg.tuwien.ac.at
- Favoritenstraße 9-11



und/oder

■ **TISS** → 186.822



Zur Lösung bieten sich an (in dieser Reihenfolge!)

- **1. Web**

- LVA-Seite, wikipedia oder google oder ...

- **2. KollegInnen**

- **3. TutorInnen**

- Diskussionsforum im TUWEL-Kurs
- Sprechstunden im Pong-Labor

- **4. zuständige Assistenten:**

- J. Unterguggenberger, B. Steiner, S. Zambanini:
evc@cg.tuwien.ac.at



- AudiMax: Mo, Di, Mi 13:15 – 14:45
- geplanter Service:
 - Streaming (ohne Gewähr)
 - Aufzeichnung (ohne Gewähr)
- “abwechselnd” Robert Sablatnig und Werner Purgathofer (je ~50%)
- letzter Vorlesungstermin (voraussichtlich): 30. Mai
- inkl. 4 Wiederholungseinheiten



■ Skriptum - Textblätter

- Kurzfassung des Wesentlichen in Deutsch
- ~4 Seiten / Doppelstunde

■ Wie, Wo, Wann?

- als PDF online zum Herunterladen in TISS
- zeitnah zur jeweiligen Vorlesungseinheit



- **Kopien der Slides (Folien)**
 - geringfügig veränderte Slides (Copyright!)
 - in Englisch
 - reichen zum Lernen alleine nicht aus!
- **Wie, Wo, Wann?**
 - als PDF online zum Herunterladen in TISS
 - zeitnah zur jeweiligen Vorlesungseinheit



■ Buch in Deutsch:

Mischwitz, Fischer, Haberäcker, Socher:

Computergrafik und Bildverarbeitung.

3. Auflage



link.springer.com/book/10.1007/978-3-8348-8323-0

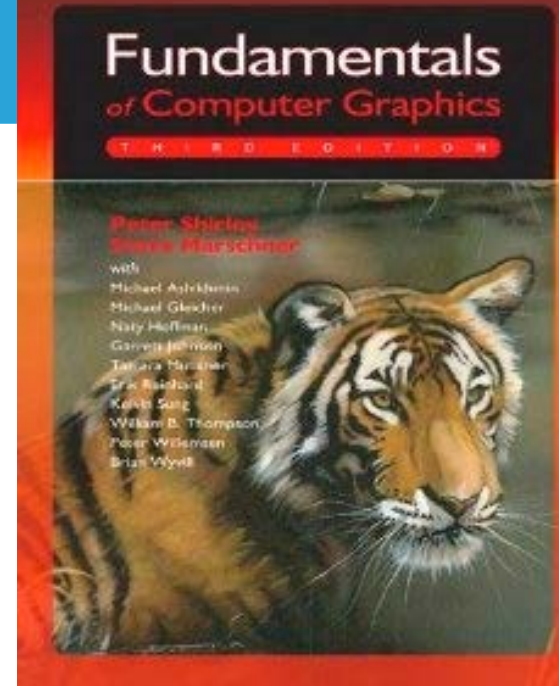


Weiterführende Unterlagen

- Engl. Buch zu Computergraphik:

Shirley, Marschner:

Fundamentals of Computer Graphics 3rd Edition

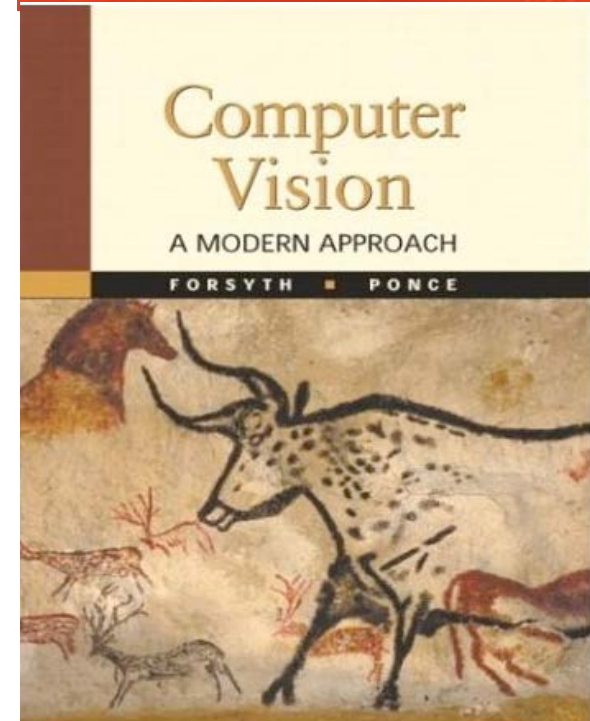


- Engl. Buch zu Computer Vision:

Richard Szeliski:

Computer Vision: A Modern Approach

szeliski.org/Book



- Details werden heute präsentiert.

ANMELDUNG zur Lehrveranstaltung:

bis 6. März in TISS

- + Eingangsbeispiel bis 15. März abgeben.
- Ohne Anmeldung keine Teilnahme!



■ Übungserfordernisse:

- 1+5 Beispiele abgeben (insgesamt 160 Punkte)
- +20 Bonuspunkte erreichbar!
- 80 Punkte (ohne Bonus) mindestens erforderlich

■ Tests:

- 2 Tests mit je 120 Punkten: 6.4., 19.6.
- 100 Punkte mindestens erforderlich
- Ersatztest Anfang Oktober ersetzt einen Test



■ Note:

- ab 200 Punkte (ohne Bonus) = positiv
- ... 340 bis 420 Punkte = sehr gut



Einführung in Visual Computing

186.822

Übungsteil

Johannes Unterguggenberger

Bernhard Steiner

Sebastian Zambanini



- Eingangsbeispiel - Bildbearbeitung

Abgabe 1:

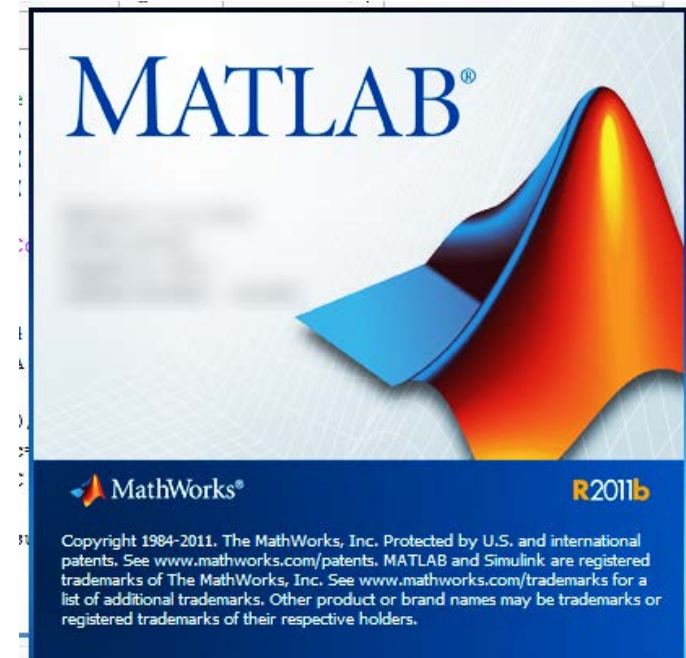
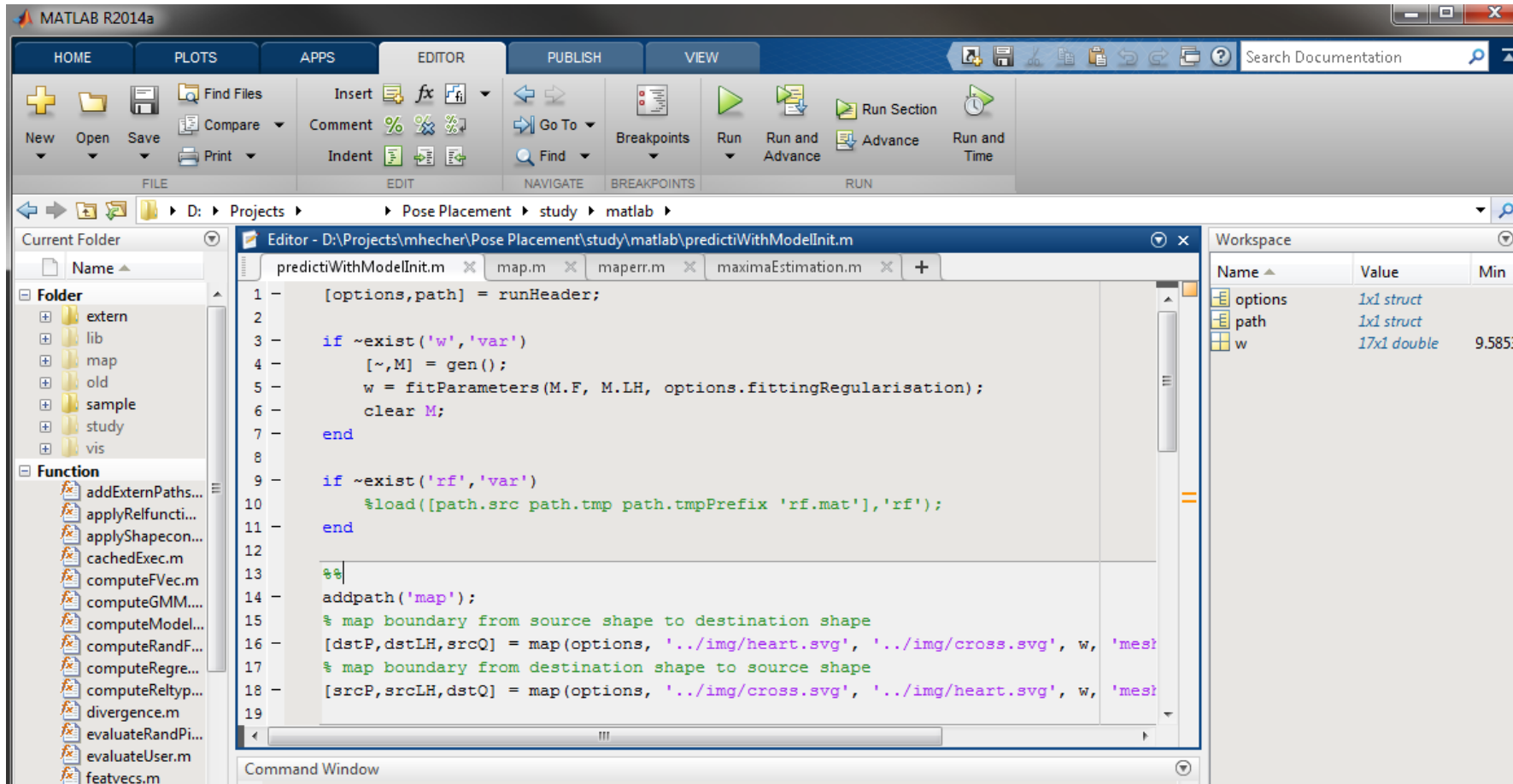
- Matlab Basics
- Kamerasensoren

Abgabe 2:

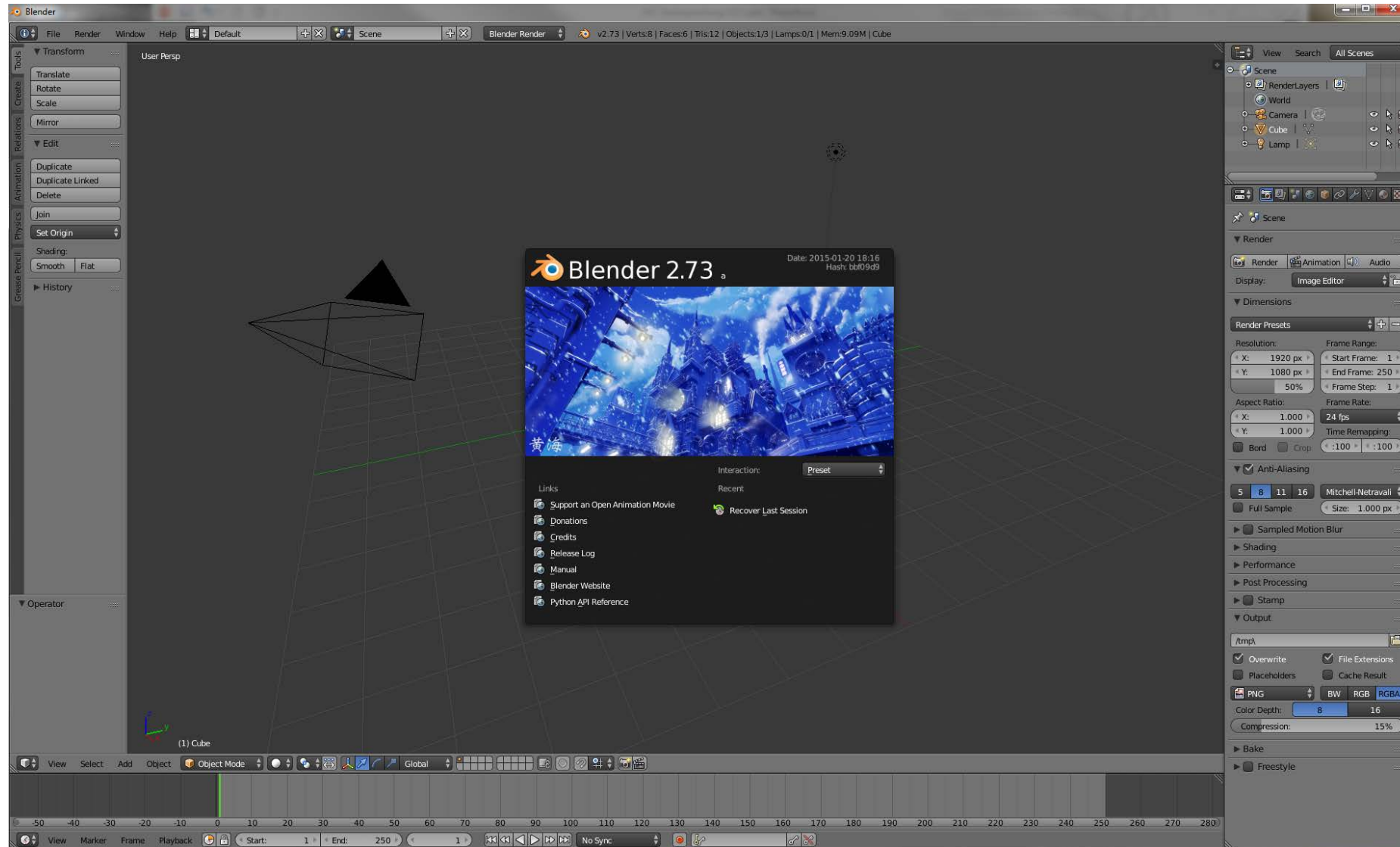
- Rekonstruktion und Modellierung
- Rasterization
- Bonusbeispiel (optional)



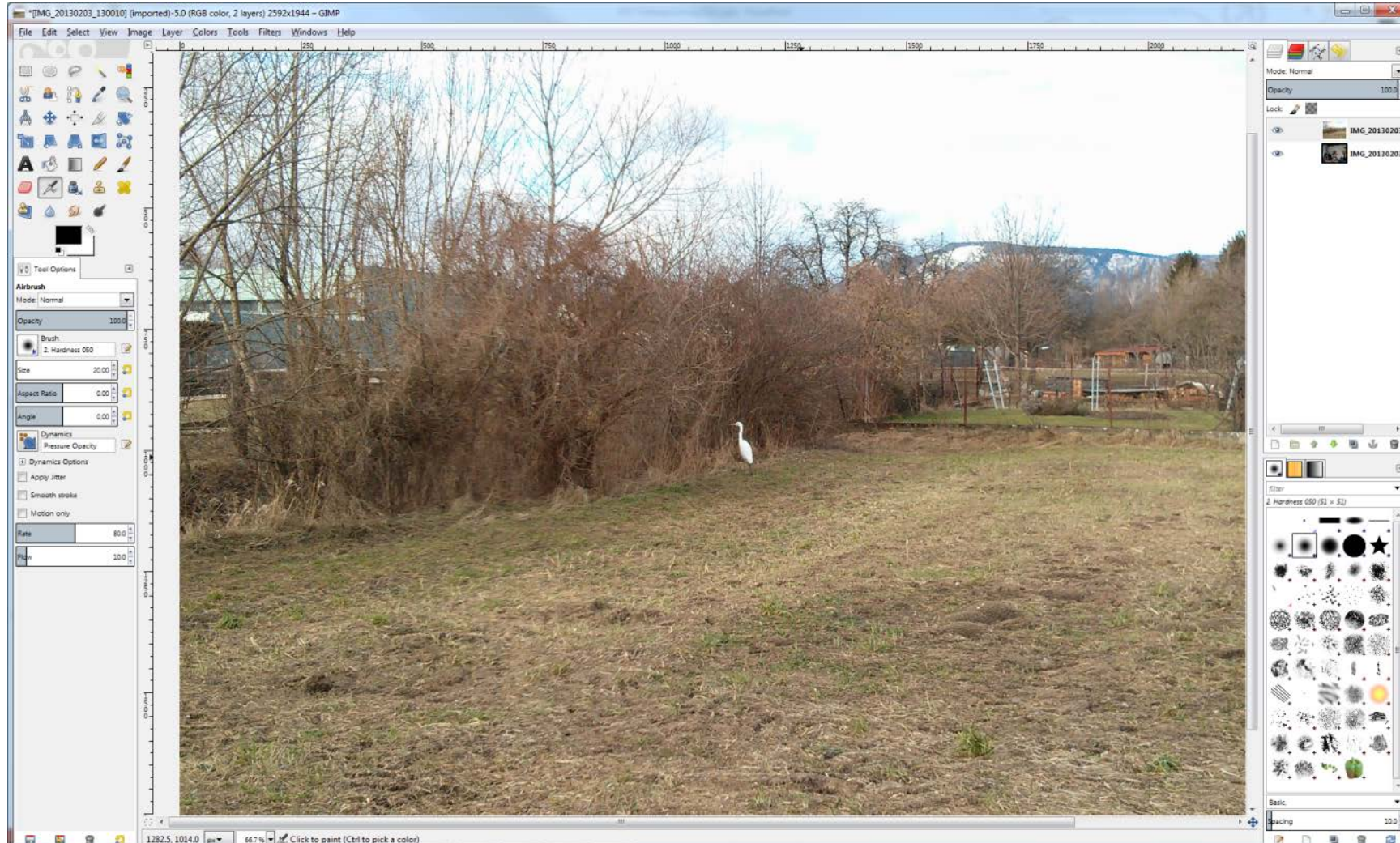
- www.sss.tuwien.ac.at/sss/
- wird im Informatiklabor verfügbar sein



- www.blender.org/download/



- www.gimp.org/downloads/



- <http://www.autodesk.com/education/free-software/remake>



- Kommunikation über das Diskussionsforum im TUWEL-Kurs
 - TISS-Forum oder andere Plattformen werden nicht betreut!

186.822 Einführung in Visual Computing (VU 5,0) 2017S



Diskussionsforum

In diesem Forum könnt ihr euch über die Übungsbeispiele austauschen und bei Unklarheiten nachfragen!
Bitte postet KEINE Codesnippets aus den Beispielen.



■ Pong-Raum

- TutorInnen bieten Sprechstunden nach der Vorlesung an
- an welchen Tagen kann man auf der LVA-Seite nachlesen:
www.cg.tuwien.ac.at/courses/EinfVisComp
- keine Anmeldung nötig



- Wichtige Termine wie Abgabefristen, Abgabegespräche und Tests finden sich auf den LVA-Seiten:
- www.cg.tuwien.ac.at/courses/EinfVisComp/
- <https://tuwel.tuwien.ac.at/course/view.php?idnumber=186822-2017S>



„Bildbearbeitung“ (verpflichtend)

- Foto mit einem Bildbearbeitungsprogramm bearbeiten



Matrikelnummer im Bild!



„Bildbearbeitung“ (verpflichtend)

- Foto mit einem Bildbearbeitungsprogramm bearbeiten

Deadline: 15. März 2017 23:55

- = verbindliche Anmeldung zur LVA!
(d.h. Sie bekommen ein Zeugnis)



Dient der Einführung in Matlab

- Verwenden der eingebauten Hilfe
- Code debuggen
- simple Bildverarbeitung
- ...

Dient der Wiederholung von

- Vektorrechnung
- Matrizenrechnung
- ...



Dient der Vorbereitung auf den Test

- Berechnungen v. Normalen (für Beleuchtung)
- Filter anwenden
- Transformationsmatrizen ...

Besteht aus 4 Teilen:

1. Basics  Matlab-Einführung **13. März 2017**
2. Triangles
3. Images
4. Transformations



- <https://tuwel.tuwien.ac.at/course/view.php?idnumber=186822-2017S>
- Login über TU-Account
- nur offen für LVA-Teilnehmer (TISS Anmeldung)
- enthält alle Angaben zu den Beispielen
 - Angaben werden während des Semesters freigeschaltet
- Anmeldung zu Abgabegesprächen
- Livestream und Aufzeichnungen der Vorlesungen
- Vorlesungsfolien und Skripten



Einführung in Visual Computing

186.822

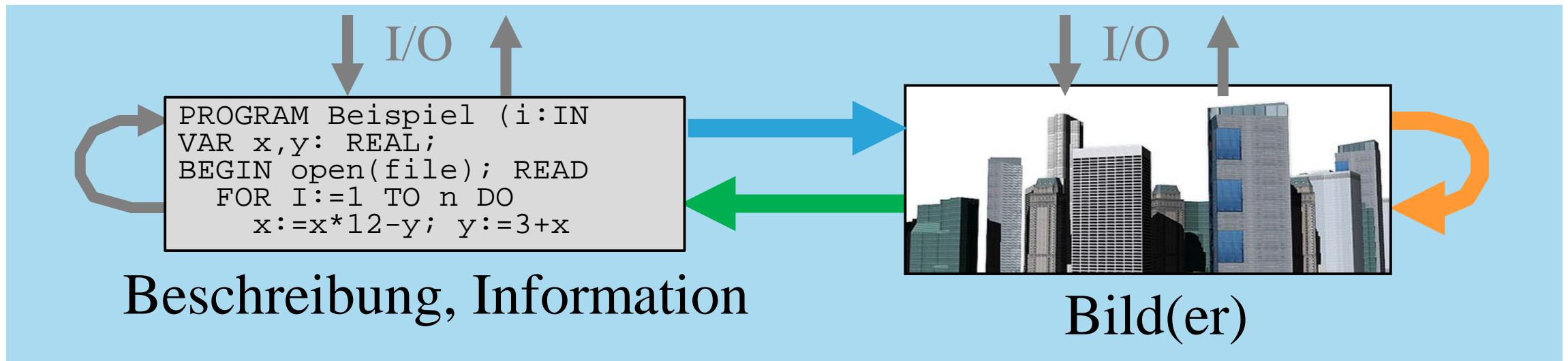
Introduction to Computer Graphics

Werner Purgathofer



Datenverarbeitung mit Bildern

- **Computergraphik:** Beschreibung → Bilder
- **Bildverarbeitung:** “schlechtes” → “besseres” Bild
- **Computer Vision:** Bilder → Beschreibung



$$CG + BV + CV = \text{„Visual Computing“}$$



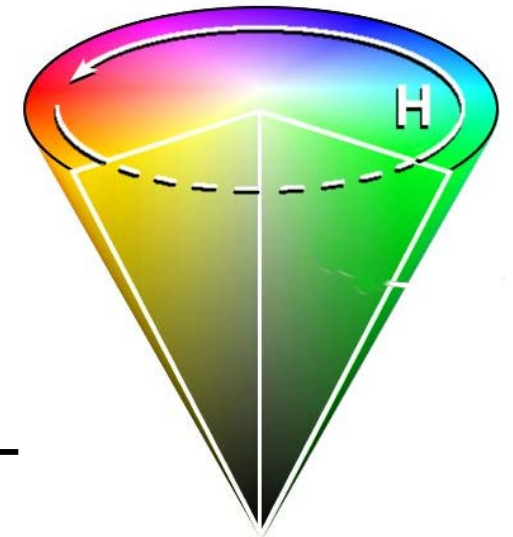
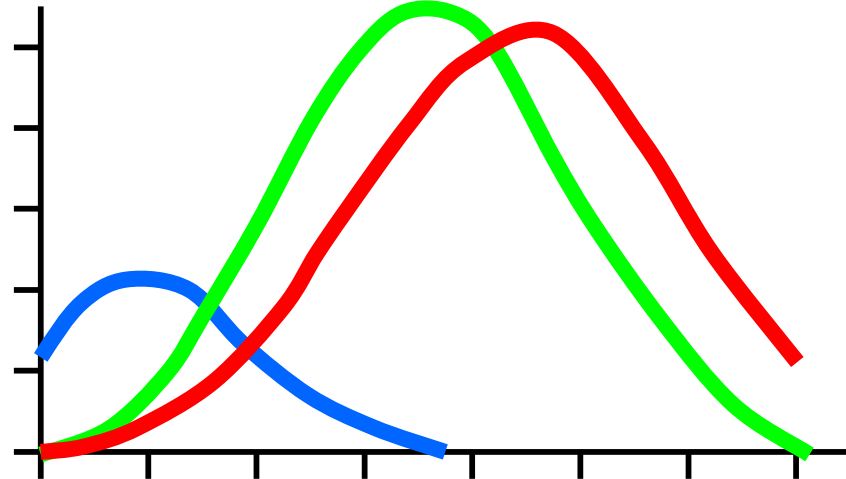
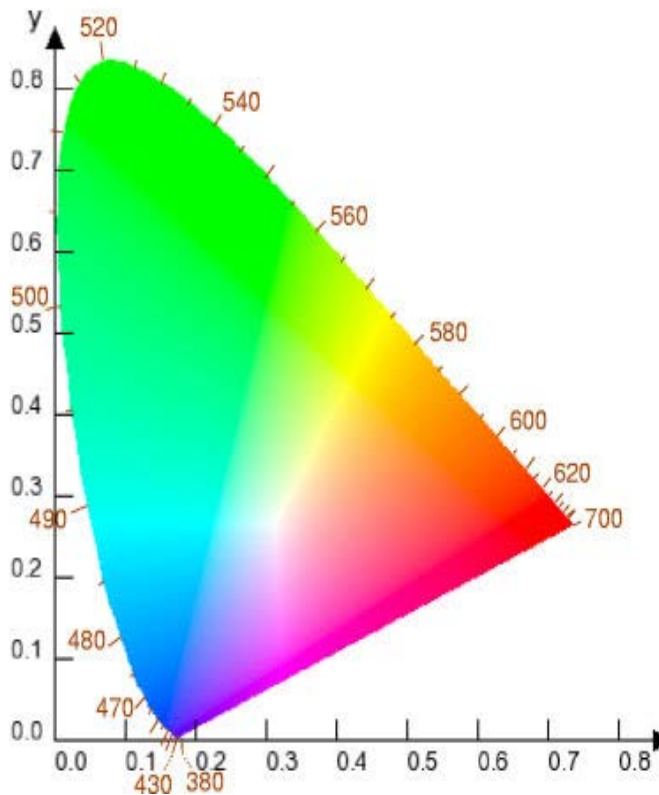
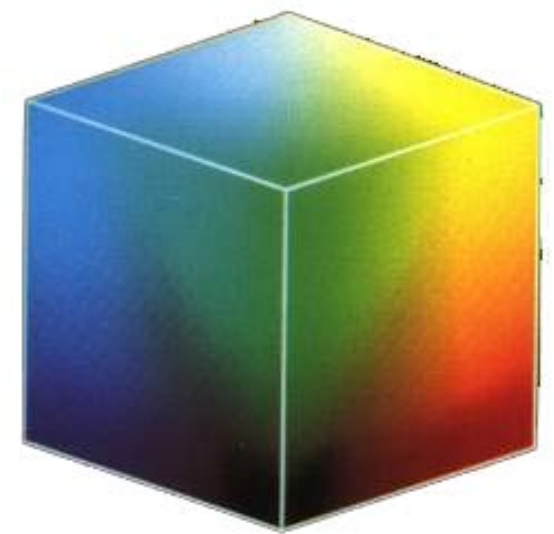
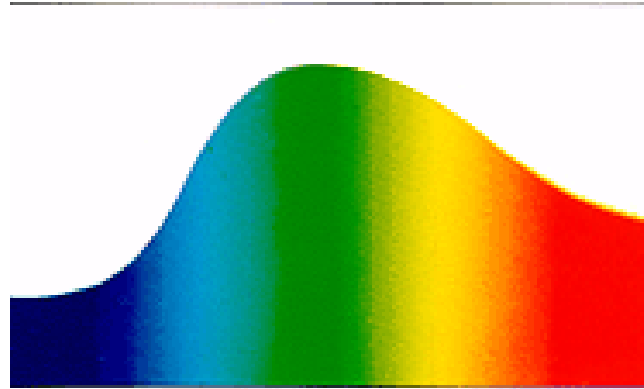
- entertainment: games, film, tv
- industrial design, architecture, landscape arch.
- marketing, advertisements
- simulators (training): cars, aircraft, spacecraft...
- perceptual rendering: security issues, design, ...
- cultural heritage, museums, learning
- science, medical visualisation
- ...



- color
- graphics primitives, rasterization
- graphics pipeline, projections, transformations
- camera definition, data structures and models
- clipping, anti-aliasing
- visibility testing, lighting + shading
- ray-tracing, global illumination
- texture maps, surface structure
- curves and surfaces



- what is color?
- how can color be described?
- color on monitor/printer



- color
- graphics primitives, rasterization
- graphics pipeline, projections, transformations
- camera definition, data structures and models
- clipping, anti-aliasing
- visibility testing, lighting + shading
- ray-tracing, global illumination
- texture maps, surface structure
- curves and surfaces

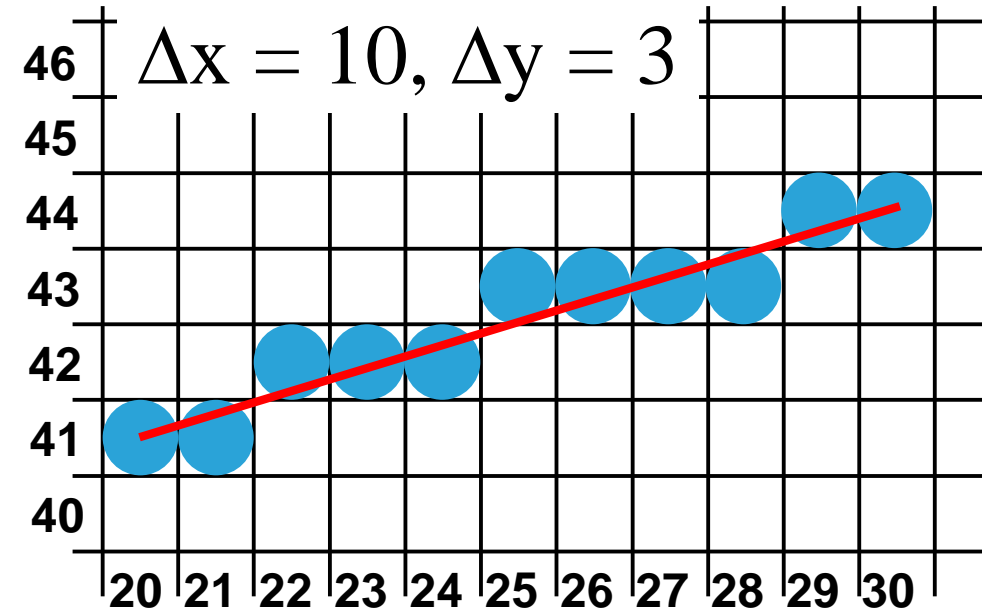
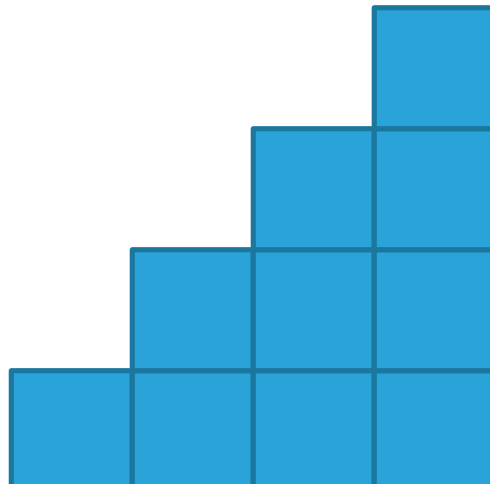


■ rasterization:

which pixels form
a line, a circle,
any primitive

■ triangle filling:

how to interpolate
inside a
triangle



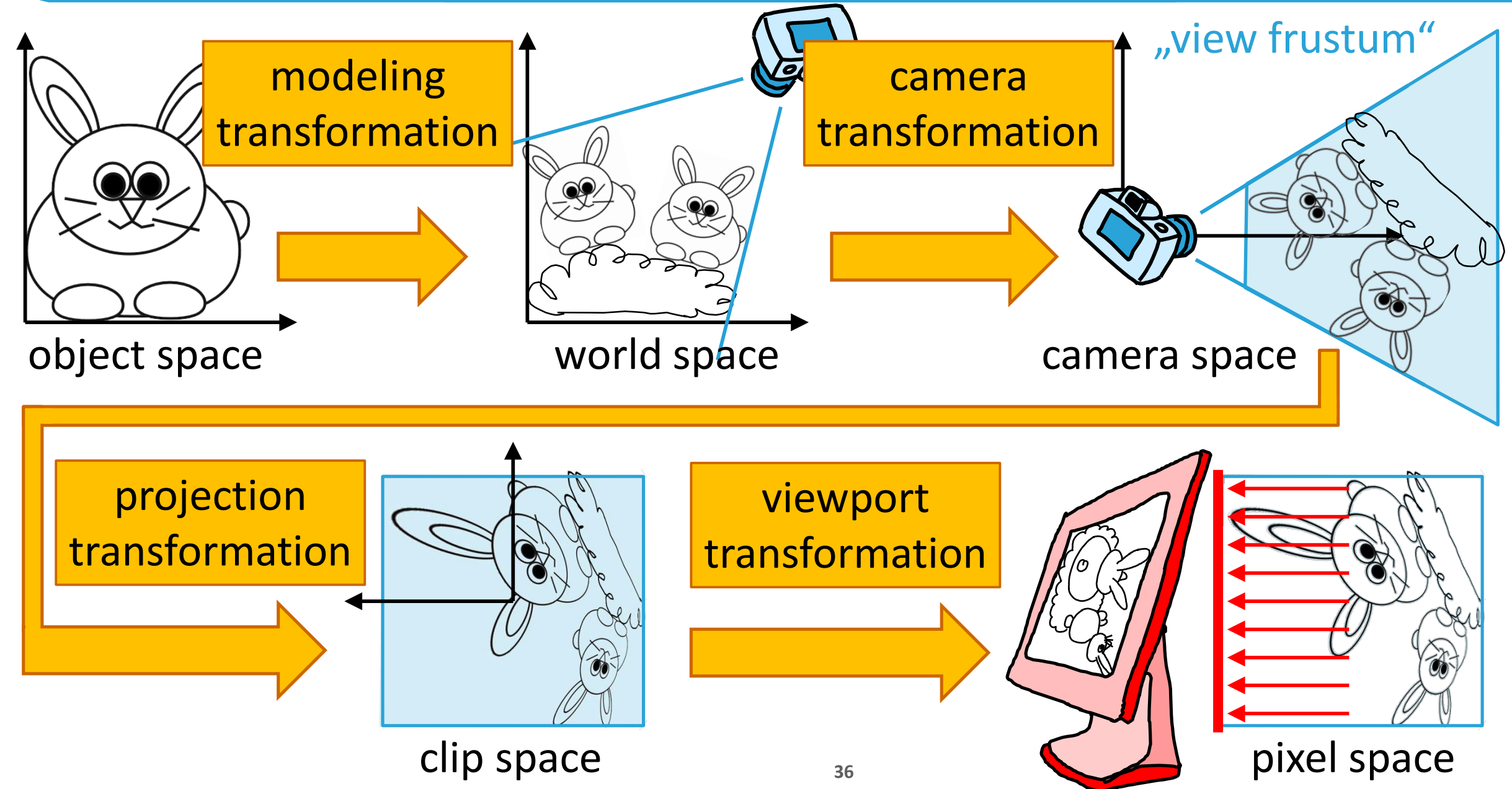
1.2	1.0	0.8	0.6	0.4	0.2	0.0	-0.2
-1.4	-1.2	-1.0	-0.8	-0.6	-0.4	-0.2	0.0
1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
1.2	1.0	0.8	0.6	0.4	0.2	0.0	-0.2
-1.2	-1.0	-0.8	-0.6	-0.4	-0.2	0.0	0.2
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.2	1.0	0.8	0.6	0.4	0.2	0.0	-0.2
-1.0	-0.8	-0.6	-0.4	-0.2	0.0	0.2	0.4
0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8
1.2	1.0	0.8	0.6	0.4	0.2	0.0	-0.2
-0.8	-0.6	-0.4	-0.2	0.0	0.2	0.4	0.6
0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
1.2	1.0	0.8	0.6	0.4	0.2	0.0	-0.2
-0.6	-0.4	-0.2	0.0	0.2	0.4	0.6	0.8
0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4



- color
- graphics primitives, rasterization
- graphics pipeline, projections, transformations
- camera definition, data structures and models
- clipping, anti-aliasing
- visibility testing, lighting + shading
- ray-tracing, global illumination
- texture maps, surface structure
- curves and surfaces

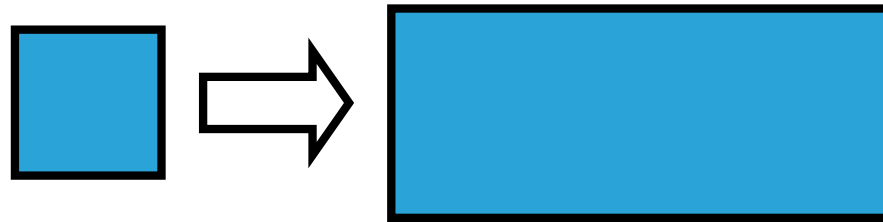


Projections in the Graphics Pipeline



- translation, rotation, scaling, ...
- simple formulas for simple transformations:

e.g. scaling



- $x' = s_x \cdot x$

- $y' = s_y \cdot y$

- general notation for all transformations: **matrices**

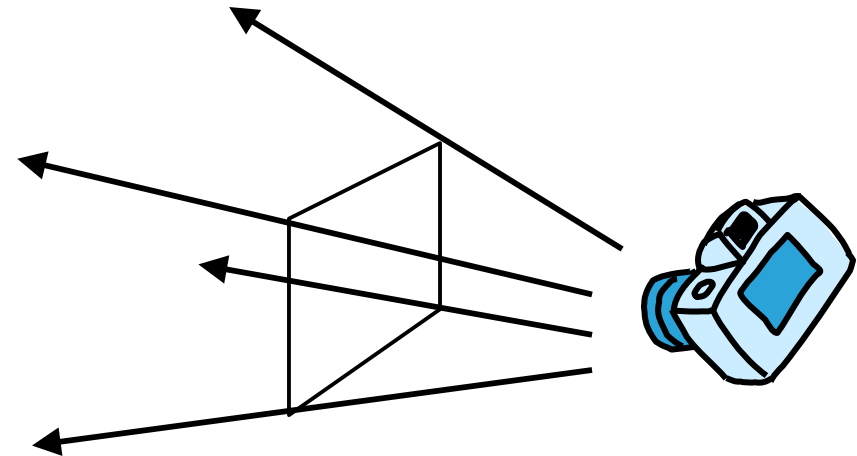
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$



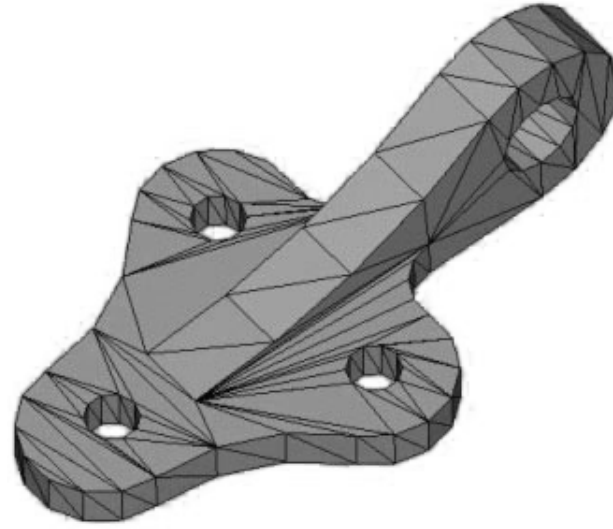
- color
- graphics primitives, rasterization
- graphics pipeline, projections, transformations
- camera definition, data structures and models
- clipping, anti-aliasing
- visibility testing, lighting + shading
- ray-tracing, global illumination
- texture maps, surface structure
- curves and surfaces



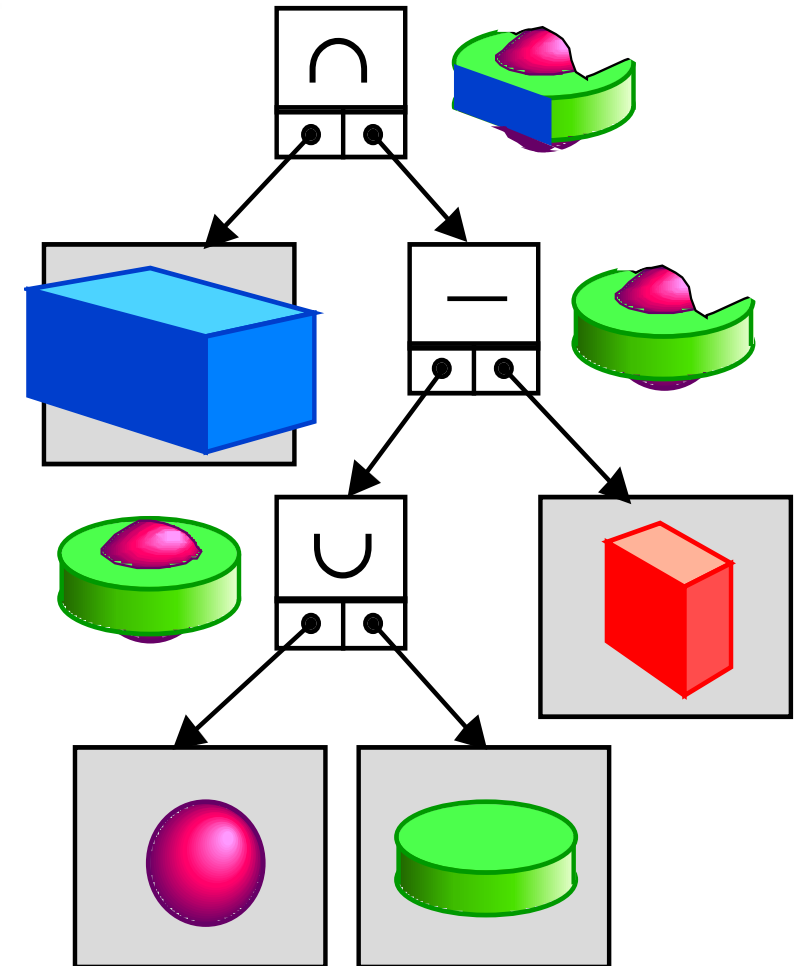
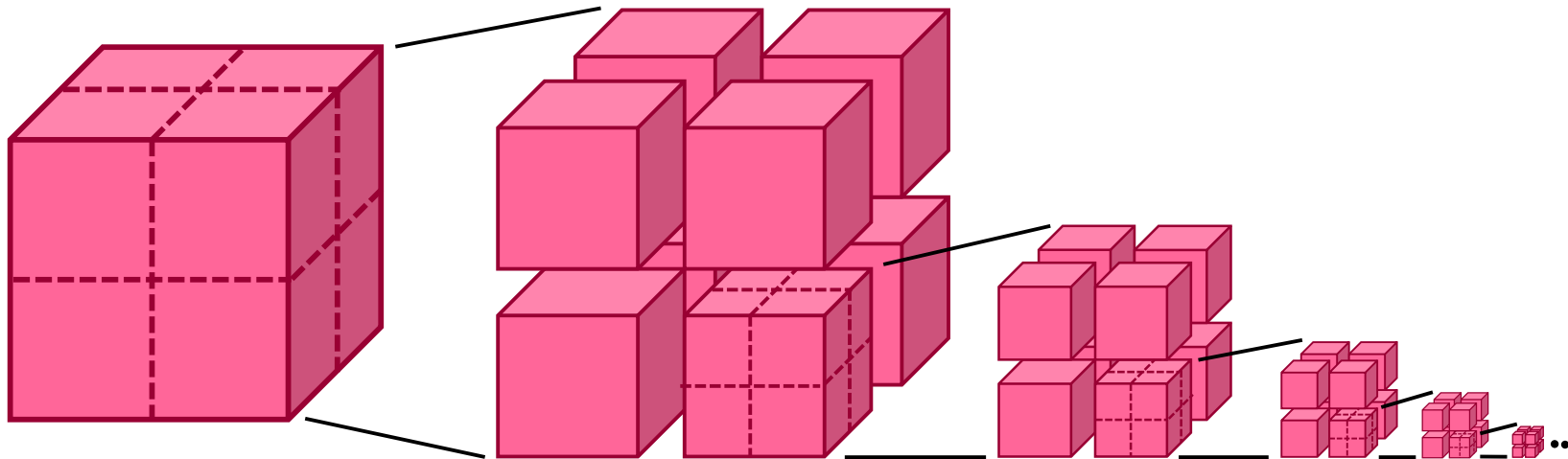
- similar to taking a photograph
- involves selection of
 - ◆ camera position
 - ◆ camera direction
 - ◆ camera orientation
 - ◆ “window” (zoom) of camera



- polygon surfaces:
general data structure



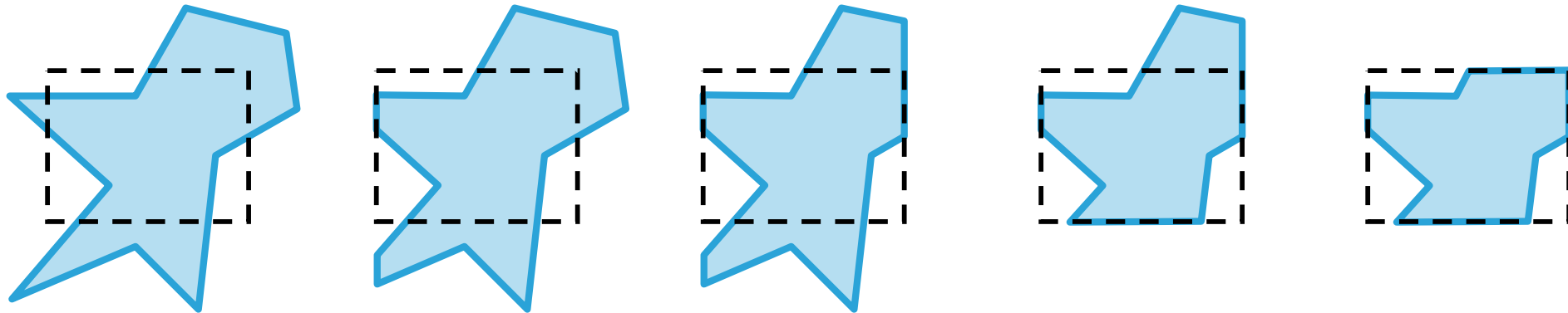
- CSG-tree, octree, ...:
special data structures



- color
- graphics primitives, rasterization
- graphics pipeline, projections, transformations
- camera definition, data structures and models
- clipping, anti-aliasing
- visibility testing, lighting + shading
- ray-tracing, global illumination
- texture maps, surface structure
- curves and surfaces

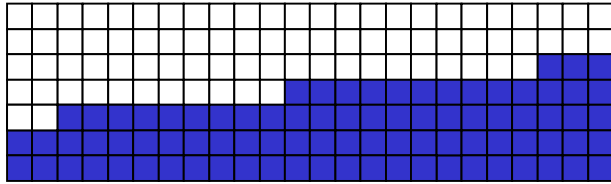


- cutting off parts that are outside window
- clipping lines = easy
- clipping polygons:

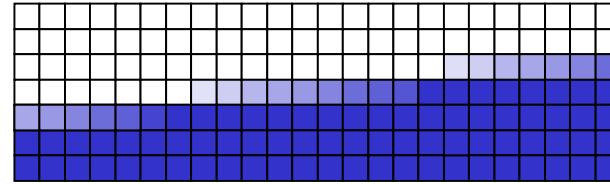


- clipping in clip-space $[0..1, 0..1, 0..1]$

- reduce discretization artifacts



aliased



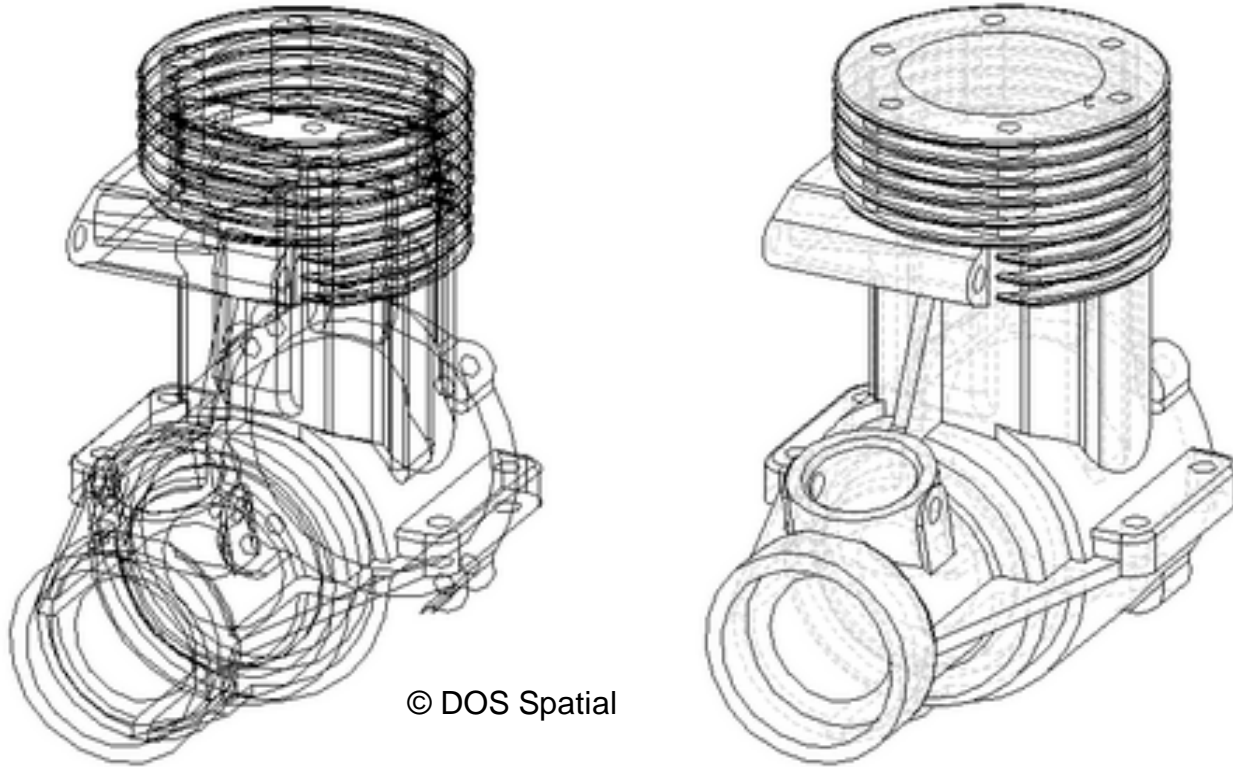
antialiased



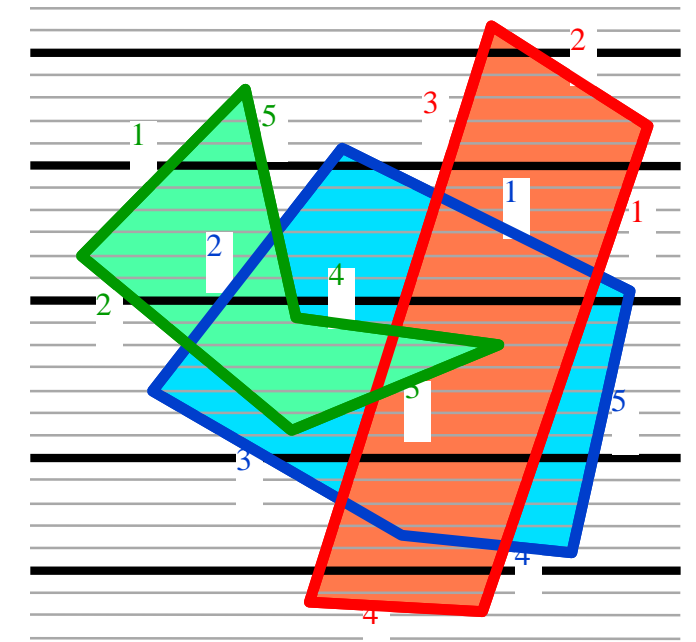
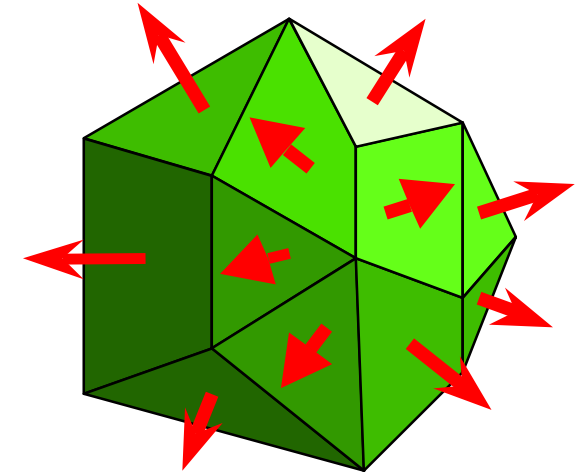
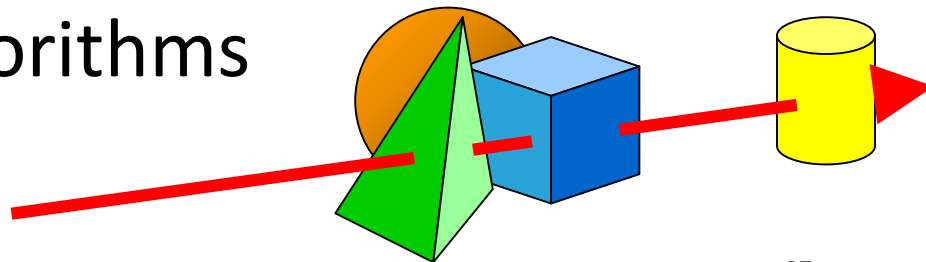
- color
- graphics primitives, rasterization
- graphics pipeline, projections, transformations
- camera definition, data structures and models
- clipping, anti-aliasing
- visibility testing, lighting + shading
- ray-tracing, global illumination
- texture maps, surface structure
- curves and surfaces



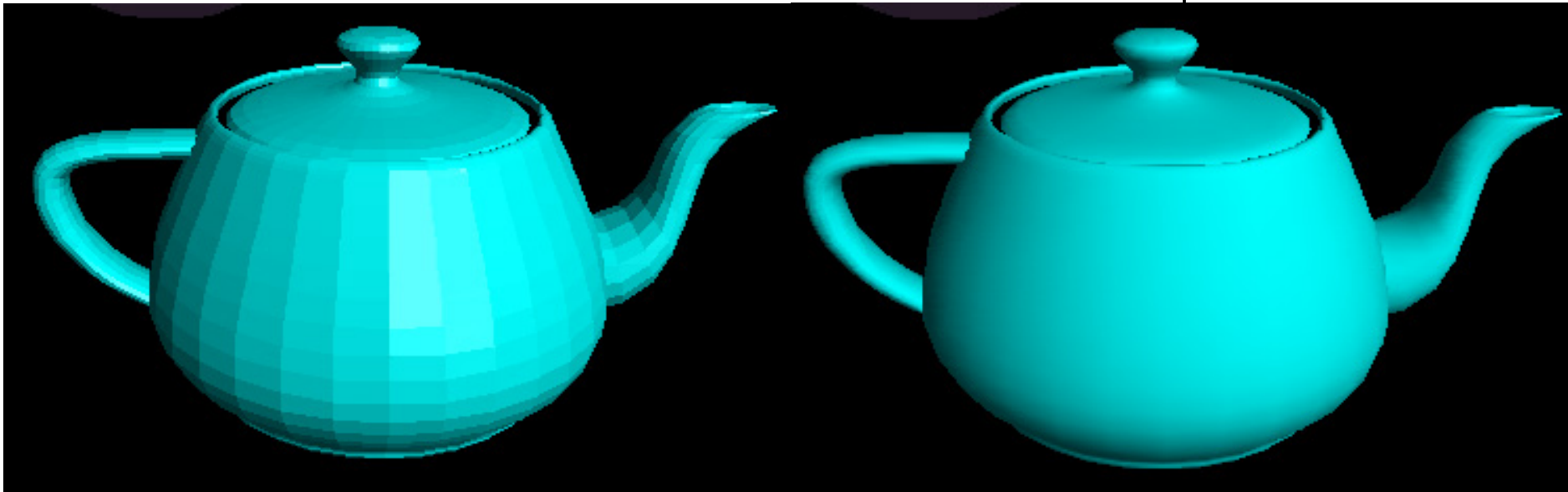
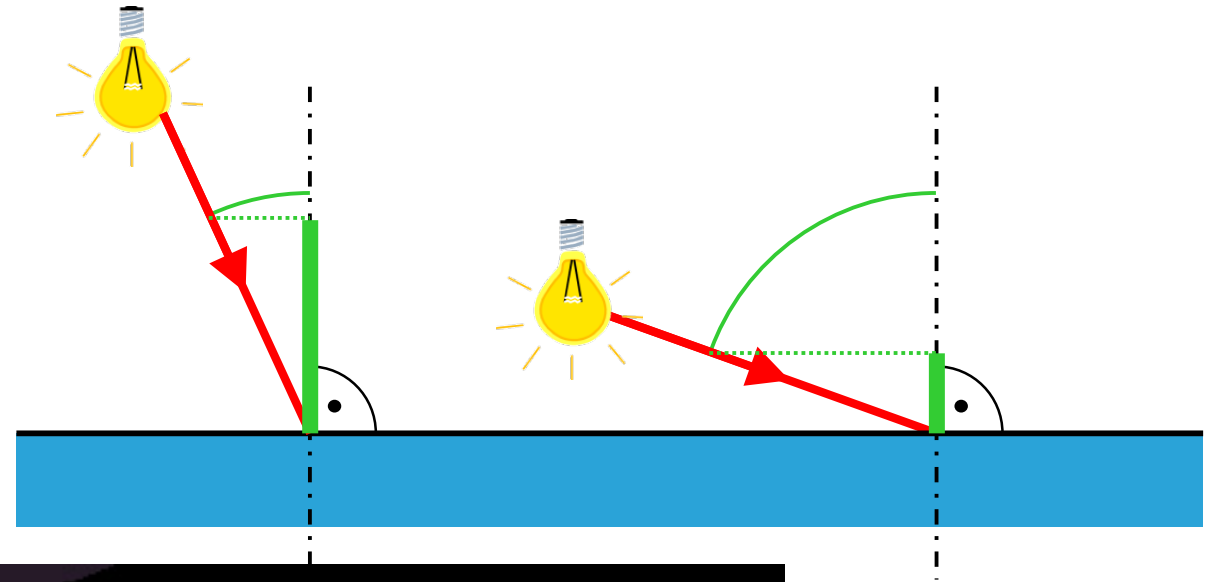
- remove hidden parts of a model



- many algorithms



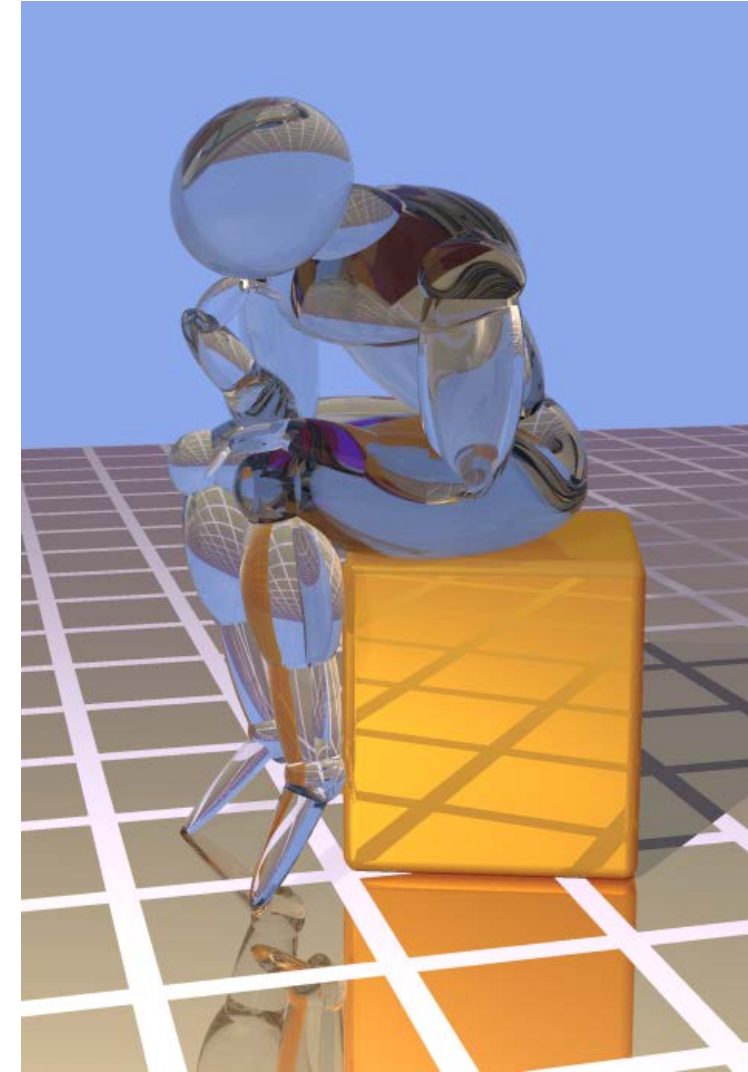
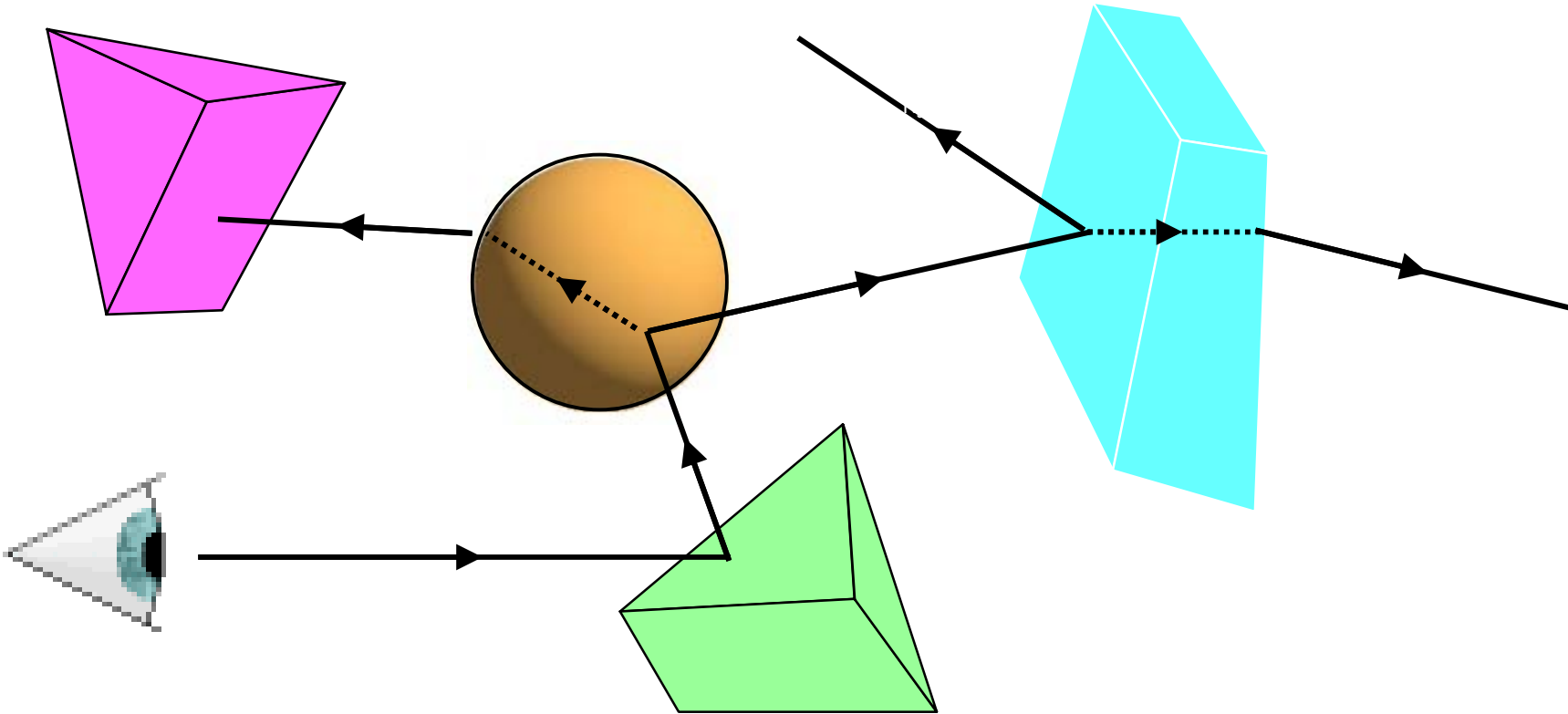
- lightsource definition
- reflection properties
- smooth shading



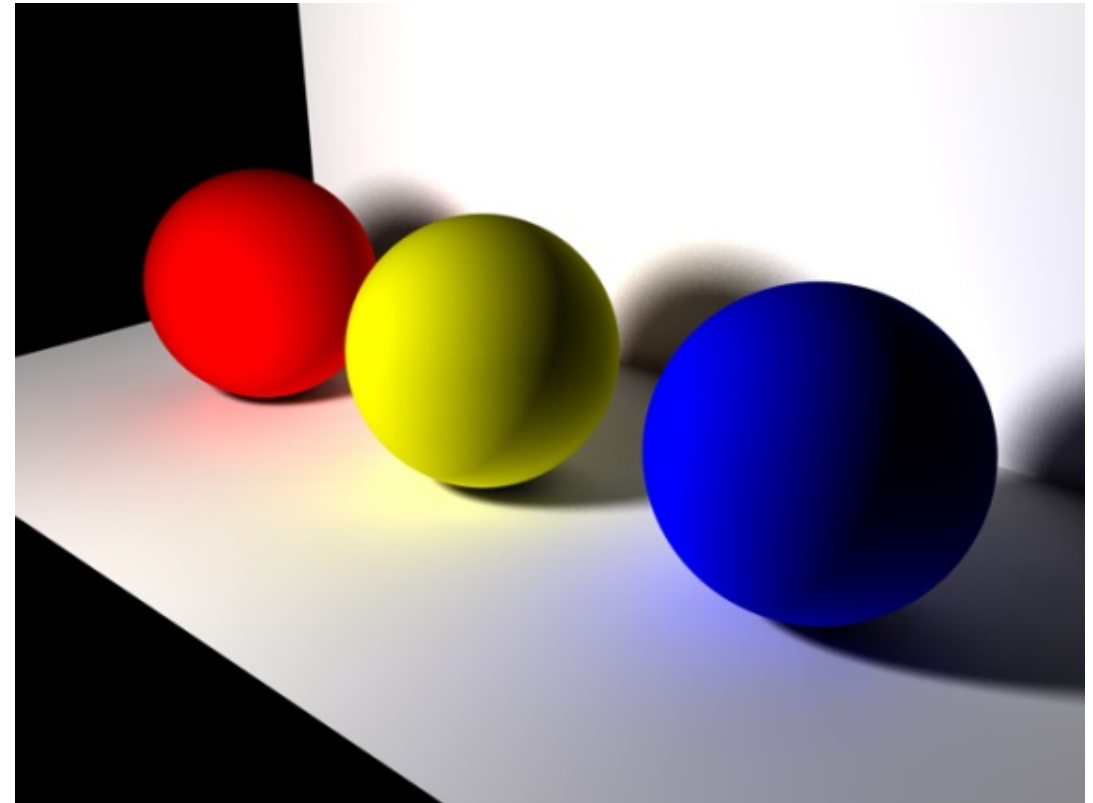
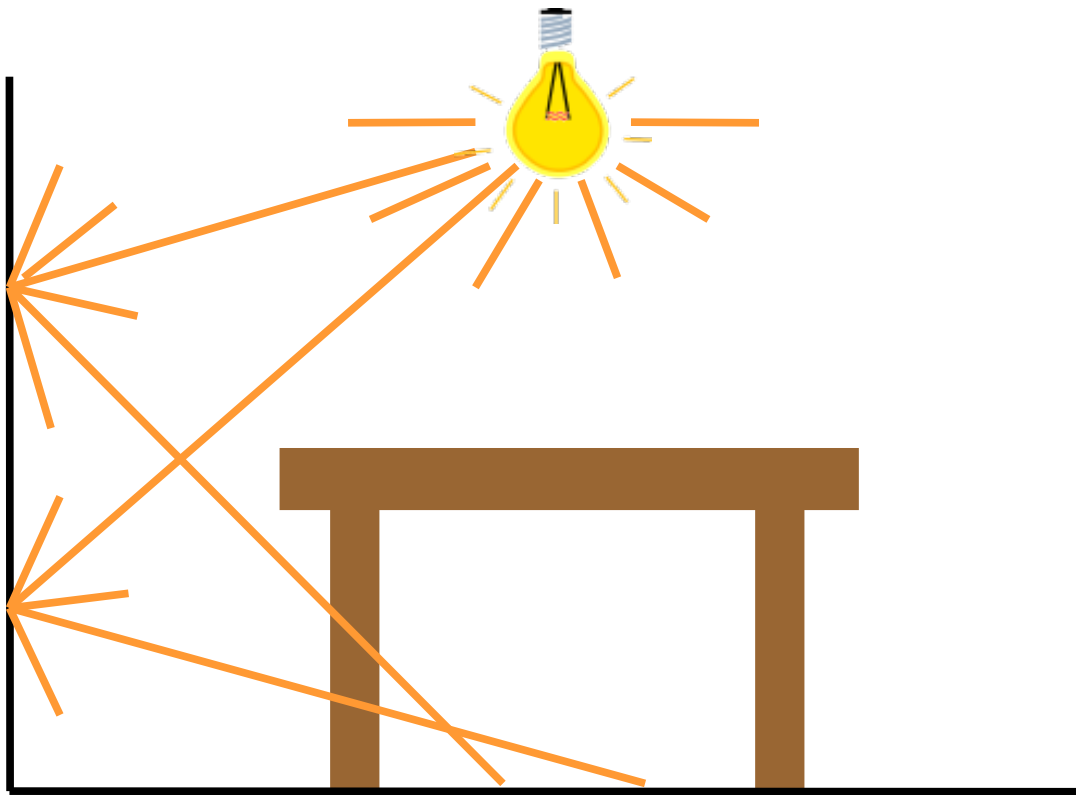
- color
- graphics primitives, rasterization
- graphics pipeline, projections, transformations
- camera definition, data structures and models
- clipping, anti-aliasing
- visibility testing, lighting + shading
- ray-tracing, global illumination
- texture maps, surface structure
- curves and surfaces



- generates realistic images by following viewing rays



- describes the physical process of light distribution in a diffuse reflecting environment

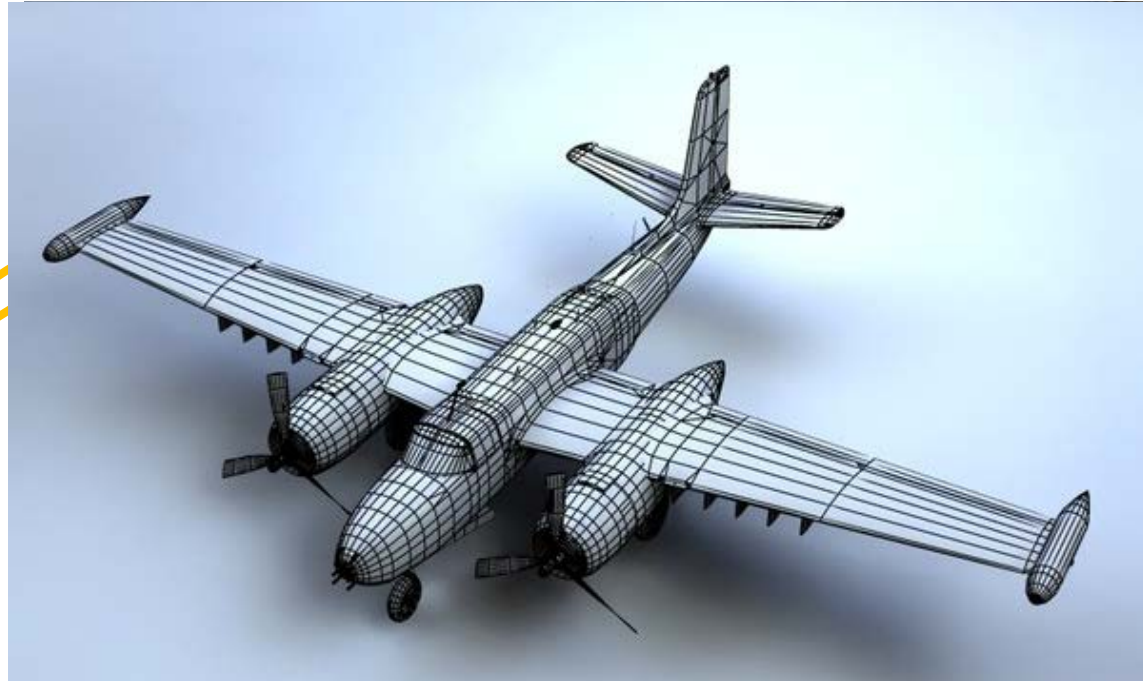
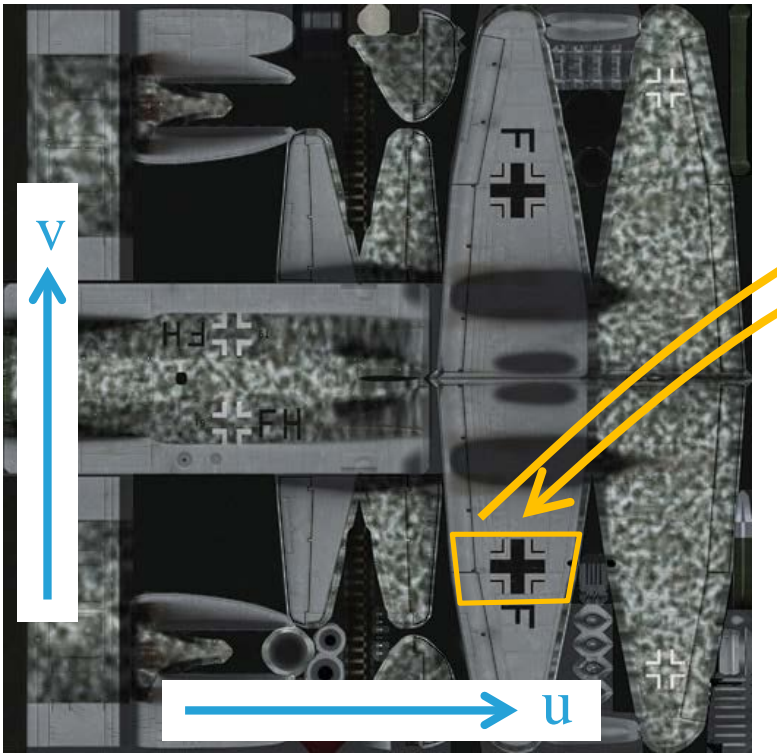


- color
- graphics primitives, rasterization
- graphics pipeline, projections, transformations
- camera definition, data structures and models
- clipping, anti-aliasing
- visibility testing, lighting + shading
- ray-tracing, global illumination
- texture maps, surface structure
- curves and surfaces



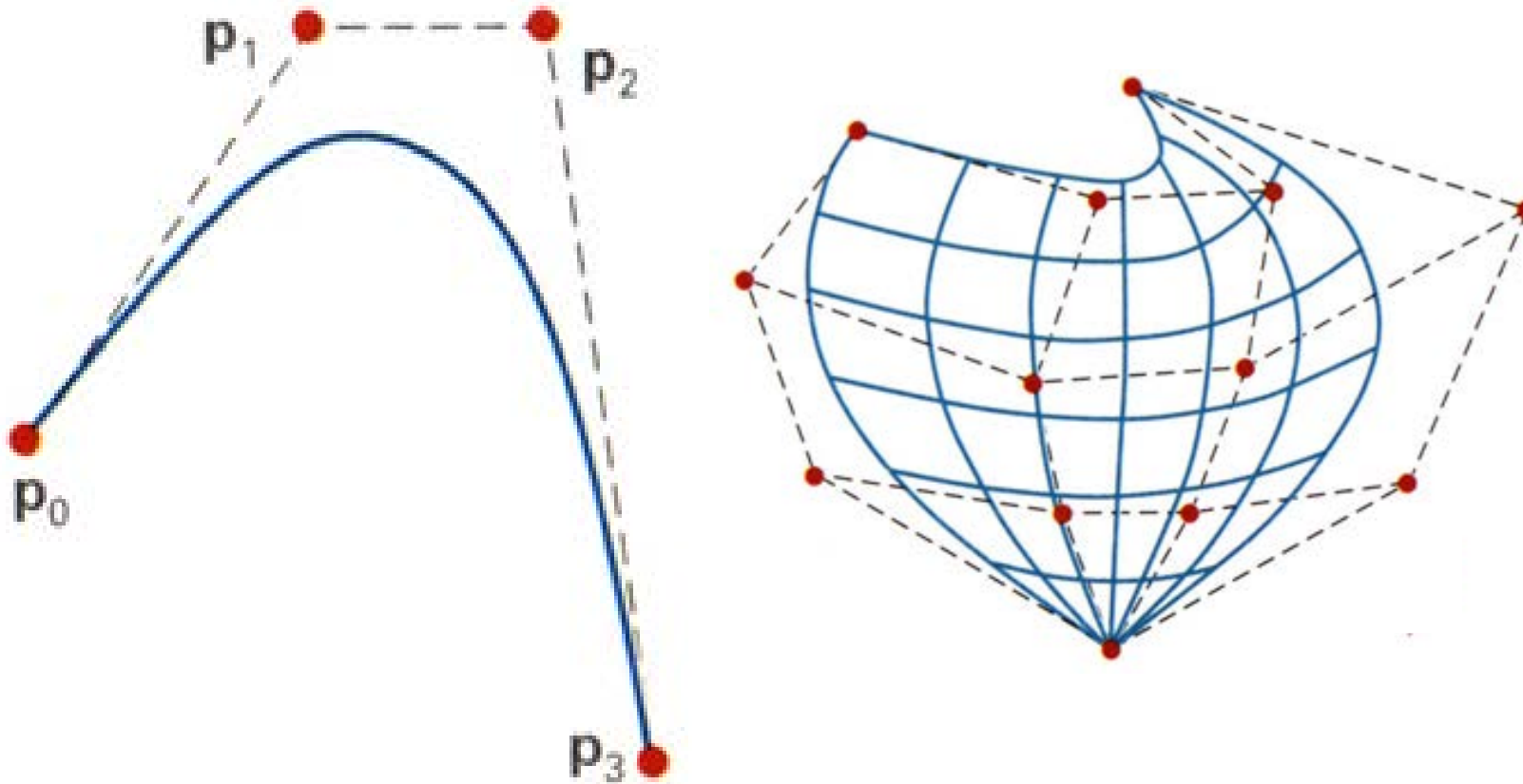
Texture Maps & Surface Structure

- adds surface details to simple models
- texture in (u,v)-space
- parametrization (mapping texture \leftrightarrow object)
- render objects with texture information



- color
- graphics primitives, rasterization
- graphics pipeline, projections, transformations
- camera definition, data structures and models
- clipping, anti-aliasing
- visibility testing, lighting + shading
- ray-tracing, global illumination
- texture maps, surface structure
- curves and surfaces

- smooth modeling with interpolating and approximating curves and surfaces



- visualization
- visual analysis
- computer animation
- virtual and augmented reality
- advanced modeling
- fractals, particle systems, ...
- realtime algorithms for graphics
- user interface design

